Information Retrieval using Word Senses: Root Sense Tagging Approach

Sang-Bum Kim, Hee-Cheol Seo and Hae-Chang Rim
Natural Language Processing Lab., Department of Computer Science and Engineering,
Korea University, Anam-dong 5 ka, SungPuk-gu,
SEOUL, 136-701, KOREA
{sbkim,hcseo,rim}@nlp.korea.ac.kr

ABSTRACT

Information retrieval using word senses is emerging as a good research challenge on semantic information retrieval. In this paper, we propose a new method using word senses in information retrieval: root sense tagging method. This method assigns coarse-grained word senses defined in WordNet to query terms and document terms by unsupervised way using co-occurrence information constructed automatically. Our sense tagger is crude, but performs consistent disambiguation by considering only the single most informative word as evidence to disambiguate the target word. We also allow multiple-sense assignment to alleviate the problem caused by incorrect disambiguation.

Experimental results on a large-scale TREC collection show that our approach to improve retrieval effectiveness is successful, while most of the previous work failed to improve performances even on small text collection. Our method also shows promising results when is combined with pseudo relevance feedback and state-of-the-art retrieval function such as BM25.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistics processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval Models, Search Process*

General Terms

Algorithms, Performance, Experimentation

Keywords

WordNet, word sense disambiguation, information retrieval, performance evaluation

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

1. INTRODUCTION

Many researchers have tried to improve retrieval performance by considering word sense[4, 11, 12, 13, 9, 8, 10]. Since natural language has its lexical ambiguity, it is predictable that text retrieval systems benefits from resolving ambiguities from all words in a given collection.

Nevertheless, it is not generally accepted that word sense disambiguation makes a meaningful contribution to the information retrieval task since most of the previous IR experiments using word senses have shown disappointing results. Sanderson compares and summarizes many of the previous works in [7], and points out some reasons for their failures, such as too skewed sense frequencies, collocation problem, inaccurate sense disambiguation, etc. Word sense disambiguation for IR tasks should be performed on all ambiguous words in a collection since we cannot know user query in advance. However, computational linguists recently report that performance of word sense disambiguation reach at most about 75% precision and recall on all word task in SENSEVAL[1] competition, although about 95% in the lexical sample task, which deals with only a few words. Thus, it is not strange that a retrieval system adopting word sense disambiguation often drops the performance by inaccurate sense disambiguation[2].

This study is motivated by some observations that sense disambiguation for crude tasks such as IR is different from traditional word sense disambiguation. First, it is arguable that fine-grained word sense disambiguation, which considers all the senses defined in a dictionary or a thesaurus, is necessary to improve retrieval performance. For example, the word "stock" has 17 different senses in the WordNet. Given that even a human can not determine the correct sense of the word in different contexts, we think that automatic disambiguation for all the fine-grained word senses certainly cause too many disambiguation errors resulting in failure of appropriate query-document matching. For this reason, *coarse-grained disambiguation* with broader senses may be preferable to *fine-grained disambiguation* for IR tasks.

It is also doubtful that highly accurate sense disambiguation is the only solution to improve IR performance. For example, disambiguation errors in a given query term would not deteriorate document-query matching if exactly the same

[1]http://www.itri.brighton.ac.uk/events/senseval/
[2]Sanderson empirically shows that 20%-30% error rate could cause effectiveness to be as bad or even worse than when ambiguity was left unresolved although the rate varied across the collections

errors also occurred in the text collection. From this point of view, we believe that *consistent disambiguation* in documents and queries according to its neighboring words is more important than traditional *accurate disambiguation*. In addition, if we want to utilize the disambiguation results as safely as possible, *flexible disambiguation*, which assigns several possible senses to a given word, is better than *strict disambiguation* assigning only the most probable sense.

In this paper, we propose a root sense tagging approach for information retrieval. Root sense means one of the 25 unique beginner senses defined in WordNet hierarchies for the noun synsets such as `person`, `act`, or `artifact`. In our approach, each noun word in a document and a query is classified into one of the candidate root senses by considering its neighboring content words. Thus, the proposed root sense tagging approach is based on *coarse-grained disambiguation* unlike the most of the previous work. When classifying a given word into one of the root senses, we select only one context word having the highest mutual information with the given word, and find the most probable candidate root sense. This enables us to perform *consistent disambiguation*, especially for many collocations or multi-word expressions. Then, all the root senses assigned to each occurrence of a term in a document are merged into a final sense unit for the index term. Using this multiple-sense assignment, i.e., *flexible disambiguation*, our approach can alleviate the problem caused by disambiguation errors. We implement it using bitwise sense field for setting one or more sense bits, and AND/OR operations for merging or matching the sense field, which requires only a small amount of additional system overhead.

2. RELATED WORKS

The most efforts on information retrieval using word senses are well discussed in [7]. In this section, we review two successful previous works closely related to our work.

[9] is the most successful work reporting 14% improvement in retrieval effectiveness using the TREC-1 category B collection. They build a word co-occurrence matrix and transformed it using SVD in order to make context clusters. Then, some of the context clusters are selected simply by finding close sense clusters using vector similarity, and assigned them to each word occurrence in query and document. This approach is a good example of *coarse-grained disambiguation* and *flexible disambiguation*. They did not use a set of fine-grained word senses that are pre-defined in an existing dictionary or a thesaurus like WordNet. They also assign a number of clusters to the word occurrences for flexible query-document matching.

Despite the good experimental results, some problems exist in their approach as pointed out in [7]. Their sense-based representation using automatically constructed global matrix is very similar to the latent semantic indexing (LSI). LSI[1, 2] is known as an alternative approach to overcome the problems caused by bag-of-word representation. However, its heavy computational cost usually makes LSI to be an unrealistic solution, especially in the case of applications dealing with large-scale text collection. Its computational cost in the retrieval on inverted index is another serious problem[5]. Their small-scale evaluation, surely due to its computational cost, and very long queries that considerably help the disambiguation are also problematic.

[10] is a recent successful work with short queries and large-scale TREC WT10G data collection. [10] empirically showed that their WSD system can significantly improve the retrieval performance. In our view, this work is categorized into *fine-grained disambiguation* and *strict disambiguation* since they use all the fine-grained senses in WordNet as well as assign a single sense to each term. Their disambiguator applies collocations, co-occurrence statistics, and prior sense frequency in a stepwise fashion. They represent documents and queries with sense vectors, and retrieve relevant documents using the traditional $tf \cdot idf$ term weighting method. However, their supervised learning method using sense-tagged SemCor corpus appears to be a problem from a practical point of view. It is also problematic that absolute performances of the baseline and the proposed system were too low to investigate the effect of sense-based text retrieval. However, we think that the most troublesome is their *strict* and *fine-grained disambiguation*, which necessarily results in many disambiguation errors.

Table 1: List of 25 WordNet unique beginners

`act`	`animal`	`artifact`
`attribute`	`body`	`cognition`
`communication`	`event`	`feeling`
`food`	`group`	`location`
`motive`	`object`	`person`
`phenomenan`	`plant`	`possession`
`process`	`quantity`	`relation`
`shape`	`state`	`substance`
`time`		

3. ROOT SENSE TAGGING APPROACH

Our proposed approach aims to improve the performance of large-scale text retrieval by conducting coarse-grained, consistent, and flexible word sense disambiguation. We use 25 root senses for the nouns in WordNet 2.0[3] shown in Table 1. In WordNet, all the noun synsets are organized into hierarchies, and each synset is part of at least one hierarchy, headed by one of the 25 root senses. For example, there are six different senses for "story" defined in WordNet, and five of them - "message", "fiction", "history", "report", "fib" - are from the same root sense of `relation`, and the sense of "floor" is from the root sense of `artifact`. Our root sense tagger classifies each noun in the documents and queries into one of the 25 root senses, which we call it *coarse-grained disambiguation*.

When classifying a given ambiguous word into one of the root senses, we first select the most informative neighboring clue word having the highest mutual information with the given word. Then, the single most probable sense among the candidate root senses for the given word is chosen according to the mutual information between the selected neighboring clue word and each candidate root sense for the given word. As a result, the tagger always assigns the same root sense to the word when the word occurs with its frequently co-occurring word, i.e., *consistent disambiguation*. The required word-word MI and word-sense MI are calculated based on the automatically constructed co-occurrence data.

For indexing and retrieval, if several different root senses are assigned to each occurrence of a specific word, they are

[3]ftp://ftp.cogsci.princeton.edu/pub/wordnet/

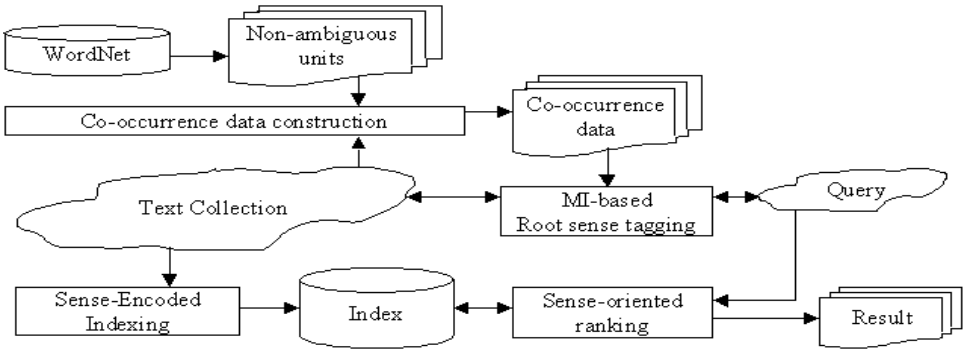

Figure 1: Root sense tagging approach for IR : System Overview

merged in the document in order to do more flexible sense-based matching between query and document. To perform the flexible matching, bitwise sense field and operations are used, which require only small amount of system overhead. Overall system architecture is shown in Figure 1.

3.1 Co-occurrence Data Construction

We regard all nouns and compound nouns having a unique root sense as *non-ambiguous units*. The list of these non-ambiguous units is the fundamental source to construct co-occurrence data between root senses and words. For example, the word "actor" has two senses including "role player" and "doer", but both senses are from the common root sense `person`. In this case, "actor" is regarded as the non-ambiguous unit for the root sense `person`. Likewise, the compound noun "computer system" has only one sense from the root sense `artifact`, and this compound noun becomes a non-ambiguous unit.

There are 101,778 non-ambiguous units in WordNet 2.0, and it is enough to build co-occurrence data between all 25 root senses and their neighboring words. Our root sense tagger is free from the requirement for a huge amount of data by considering only 25 root senses, while traditional fine-grained sense disambiguators usually face the difficulty of data sparseness to deal with hundreds or thousands of pre-defined word senses.

Given raw text collection and a list of non-ambiguous units, co-occurrence information has been extracted from each document by the following steps:

1. Assign a root sense to each non-ambiguous noun in the document.
2. Assign a root sense to each second noun of non-ambiguous compound nouns in the document.
3. Even if any noun tagged in step 2 occurs alone in other position, assign the same root sense in step 2.
4. For each sense-assigned noun in the document, extract all (*context word*, *sense*) pairs within a predefined window.
5. Extract all (*word*, *word*) pairs.

For instance, if "computer system" - a non-ambiguous compound noun having the unique root sense "artifact" - occurs in a given document, the second noun "system" is labeled as "artifact" in step 2, and all the occurrences of "system", even not followed by "computer", in the document are also labeled as `artifact`. It is based on one-sense-per-collocation and one-sense-per-discourse assumption in [13]. The *context word* in step 4 indicates k nearest content words on the left side and those on the right side of the sense-assigned noun. In our experiments, only nouns, verbs, and adjectives are considered to be content words, and a window parameter k is arbitrarily set to 2. (*word*,*word*) pairs in step 5 are also extracted within the same windows.

All extracted (*context word*,*sense*) and (*word*,*word*) pairs from the collection are compiled into global co-occurrence data. Using this data, we can assign all the remaining ambiguous units in the indexing phase. Table 2 shows the example of mutual information between three example words and their most frequently co-occurring top three root senses.

Table 2: Top 3 MI-valued root senses for word "build", "famous", and "last"

"build"	MI	"famous"	MI	"last"	MI
`artifact`	22.75	`person`	3.59	`time`	55.51
`body`	8.27	`object`	3.59	`person`	17.19
`group`	6.42	`state`	0.77	`act`	13.51
:	:	:	:	:	:

3.2 MI-based Root Sense Tagging

When we preprocess the documents for indexing, all non-ambiguous unit nouns are sense tagged with their root senses again in the same way as described in the section 3.1. In this section, we describe the MI-based root sense tagging method, which automatically assigns root senses to ambiguous words using global co-occurrence data constructed by the steps described in the previous section.

Our method is a very simplified version of existing word sense disambiguation methods:

- First, select the most related context word $c(w)$ to ambiguous word w among the context words by mutual information as follows:

$$c(w) = \text{argmax}_{c_i \in cw(w)} MI(cw, c_i) \quad (1)$$

- Second, find the highest MI-valued candidate root sense $s(w)$ with the selected $c(w)$ in the first step as follows:

$$s(w) = \text{argmax}_{s_i \in cs(w)} MI(c(w), s_i) \quad (2)$$

Table 3: Examples of disambiguation for "system" and "interest"

"interest"
... and bay has been designated site of special scientific *interest* ... (`cognition`)
... nations with very different *interests* could never reach consensus ... (`cognition`)
... many hours of discussion between various *interests* represented on it ... (`cognition`)
... government, which has not paid *interest* on its bank debt since ... (`possession`)
... *interest* margin is 55 basis points over Libor ... (`possession`)
... related to changes in poll tax and *interest* rates ... (`possession`)
"system"
... makes other automation control components and *systems*, such as vision recognition ... (`artifact`)
... TCS stands for traction-control *system* ... (`artifact`)
... if study's plan for opening up auction *system* is adopted ... (`body`)
... enable Poland to maintain the open import *system* that we now enjoy ... (`body`)
... *system* is changing slowly but be suspicious ... (`cognition`)
... It made sense to change *system* slightly so that ... (`cognition`)

where $cw(w)$ indicates the set of the context words for w defined in the previous section, and $cs(w)$ indicates the set of the candidate root senses of w defined in WordNet.

Sometimes, there is no co-occurrence data for calculating mutual information to find $c(w)$ or $s(w)$. In these cases, we set $s(w)$ to `null`. In addition, we set $s(w)$ to `unk` for unknown words not defined in WordNet. We consider only the single most related word as the contextual evidence in order to perform consistent disambiguation especially for collocations and multi-word expressions.

For example, "interest rate" is a compound noun having the unique root sense `possession`, so the second word "rate" is classified into `possession` and the first noun "interest" should be automatically classified using co-occurrence information. In this case, "rate" is selected as $c(interest)$ in most cases since "interest" and "rate" occur with each other very frequently. Once "rate" is selected for $c(interest)$, the root sense `possession` is undoubtedly selected as $s(interest)$ because the word "rate" has the highest MI value with the root sense `possession` among the candidate root senses of "interest". The same result is obtained in the case of "..rate of interest.." or even in contexts such as "..interest in low rate accounts..".

Table 3 shows examples of disambiguation for "interest" and "system". The examples for "interest" shows that adjective "special" or "different" usually co-occur with the nouns belonging to `cognition` among the possible candidate senses of "interest", while "pay" or "margin" co-occur with the `possession` nouns. The examples for "system" in this table show another characteristic of our disambiguation approach. The word "system" is one of the vague words to clearly disambiguate the meaning in the context. In WordNet, "system" has 9 different fine-grained senses, and 5 different root senses: `artifact`, `cognition`, `body`, `substance` and `attribute`. One may think that disambiguation of "system" would be useless even to human in understanding the meaning of text. Our root sense tagger also gives somewhat strange results as shown in this table. However, we can expect the phrasal indexing effect or proximity-based ranking by specializing the common word "system" according to whether it is co-occurred with "control", "open", or "change". If "system" occurs with "control" in a query, the documents containing "system" near "control" have higher relevance scores because "system" in both the query and the documents are probably classified into `artifact`.

If we consider more neighboring words in the disambiguation phase, more accurate disambiguation may be possible. However, we may also encounter a bad situation that different root senses for a word are assigned in the query and document even if they occur with the same word. For example, the word "system" co-occurring with "control" in a document is tagged as `artifact`, but `body` may be tagged to "system" immediately followed by "control", resulting in a sense-mismatch between the query and the document. This case is less desirable than the situation where the same incorrect root sense is assigned to both terms in a query and a document, which at least results in a sense-match. This is why we decide to perform *consistent disambiguation* in a primitive way using only the single context word, even though we sacrifice the possibility of performing more accurate disambiguation.

3.3 Indexing and Retrieval

Our indexing and retrieval strategy is basically based on bitwise sense field and operations. We add additional 26-bit sense field to each term posting element in index. Assigned root senses for each unique term are encoded by setting proper sense bit among the 25 bits in the sense field. The remaining 1 bit is used for `unk` assigned to unknown words. If $s(w)$ is set to `null` or w is not a noun, all the bits in the sense field are set to 0.

Unlike to the traditional bag-of-word inverted indexing approach, we must consider the following two situations caused by employing sense field. First, several different root senses may be assigned to the same word within a document according to their different contexts in the document. Second, our sense tagger assigns root senses to only nouns, but a verb with the same indexing keyword form may exist in the document. For example, our root sense tagger may assign two different root senses to the noun "certification" occurring twice in a document, but does not assign any sense to the verb "certify". In this case, we simply merge all the sense fields by the bitwise-OR operation, and take the merged sense field for its final sense field as shown in Figure 2. By allowing multiple-sense assignment, we can avoid the problem caused by inaccurate strict disambiguation and exponential increase of the index size[4].

[4]If a traditional bag-of-words indexing system defines two data fields (i.e. term identifier and within-frequency) for

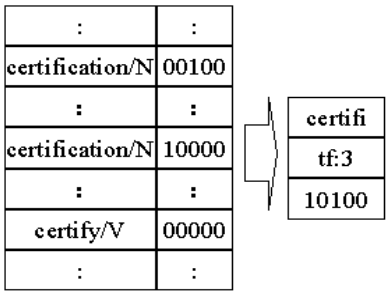

Figure 2: Example of sense merging

Many previous approaches[2, 9, 10] require both a sense-based index and a term-based index, retrieve documents based on each index, and finally combine the results. This approach has several disadvantages including serious computational complexity and ad hoc heuristics of combining the results. Moreover, if we employ their sense-based matching with our coarse-grained sense tagging, our system will return a huge number of the documents. For example, our system will retrieve most of the documents in a collection for the user query "car" because `artifact` is tagged to the "car", and the most of the documents has something tagged by `artifact`.

For this reason, we propose a *sense-oriented term weighting method* to rank documents considering word senses. Our sense-oriented term weighting just maintains the traditional term-based index, and artificially transforms term weight using sense weight sw calculated by refering to the sense field of each term posting in the retrieval phase. Sense weight sw_{ij} for term t_i in document d_j is defined as follows:

$$sw_{ij} = 1 + \alpha \cdot q(dsf_{ij}, qsf_i) \quad (3)$$

where dsf_{ij} and qsf_i indicate the sense field of term t_i in document d_j and query respectively. Here, α is a parameter controlling the impact of sense-matching result by q-function. Sense-matching function q is defined as follows:

$$q(dsf_{ij}, qsf_i) = \begin{cases} 0 & \text{if } (dsf_{ij}{=}0) \text{ or } (qsf_i{=}0) \\ 1 & \text{else if } (dsf_{ij} \ \& \ qsf_i) \neq 0 \\ -1 & otherwise \end{cases} \quad (4)$$

Sense-matching function q returns 0 if a root sense is not assigned to a term either in a query or document. Otherwise, the function returns a positive sign or a negative sign. It returns either a positive sign if there is any common bit setting as 1 or a negative sign if there is no common bit setting as 1.

For the ranking, this sense weight is multiplied by the original term weight computed in the traditional way such as $tf \cdot idf$. In other words, we boost the original term weight if the sense of a word in a query is contained in the sense field of the word in a document. Otherwise, we take the original weight itself, or cut it down when the sense of the word in a query is not contained in the sense field of the word in the document.

each posting element, our strategy requires only about 30% increase in the original index size.

4. EXPERIMENTAL RESULTS

4.1 Data and Evaluation Methodologies

In this section, we have carried out a large-scale evaluation of the proposed root sense tagging approach for information retrieval. All the experiments were conducted using two different document collections and two different query sets. For the document collections, 210,157 documents of Financial Times collection in TREC CD vol.4 and 127,742 documents of LA Times collection in vol.5 were used. For the query set, we used TREC 7(351-400) and TREC 8(401-450) queries.

One way to examine the usefulness of our root sense tagging approach is to check the boolean search performances, that is, to retrieve only the documents containing the query term whose sense is also same as the one assigned in the query. However, since traditional weighting methods using tf and idf are much more popular than boolean retrieval, the performance evaluation using simple boolean retrieval may be unnatural. Moreover, the ranking performance can not be measured by the boolean search evaluation. For this reason, we rank documents using tf and idf, and evaluate our root sense tagging approach by multiplying the sense weight sw defined in Eq.(3) to the baseline term weight.

We use the following three baseline term weighting methods:

- W1 : simple idf weighting
- W2 : $tf \cdot idf$ weighting
- W3 : $(1 + log(tf)) \cdot idf$ weighting

Although there are several excellent term weighting heuristics such as length normalization or smoothing term frequencies, we did not use those methods to clearly investigate the behavior of sense-oriented term weighting. Exceptionally, we have tested logarithm heuristics for term frequency in W3, which is well-known heuristics to improve IR performance, to check whether our proposed sense-oriented term weight is still valid with the heuristics. For the evaluation measures, we used two well-known measures including 11 points non-interpolated average precision and precision at 10 documents (p@10).

4.2 Text Retrieval with Root Sense Tagging

Table 4 shows the 11 points non-interpolated average precisions of the three baseline retrieval methods and their sense-oriented term weighting versions(+sense). In this table, "ft" and "la" represent the collection names, "7" and "8" mean the query set, and "t" and "d" indicate the title (short) and description (long) queries in those query sets. This table shows that our root sense tagging approach consistently contributes to the retrieval performances on both the different collections and the different query sets. In the idf-only weighting (W1) experiment, we achieve 4.42% point and 12.83% point improvements in average for the short and long query experiments. Sense weight parameter α in Eq.(3) is set to arbitrary value 0.5. It is obvious that the performances might worsen in the long query experiment if there are many inaccurate or inconsistent root sense assignments. However, we can obtain more improvements in the long queries experiments. Therefore, we can claim that the MI-based root sense tagging successfully specialize the terms using root senses.

Table 4: Performances measured by non-interpolated 11 points average precisions

Exp	W1	W1+sense	ΔW1	W2	W2+sense	ΔW2	W3	W3+sense	ΔW3
ft7t	0.1371	0.1447	5.54%	0.1092	0.1140	4.40%	0.1772	0.1805	1.86%
ft8t	0.1788	0.1868	4.47%	0.1262	0.1292	2.38%	0.2243	0.2265	0.98%
la7t	0.1208	0.1233	2.07%	0.1010	0.1090	7.92%	0.1752	0.1784	1.83%
la8t	0.1221	0.1287	5.41%	0.0834	0.0848	1.68%	0.1809	0.1726	-4.59%
avg	0.1397	**0.1459**	4.42%	0.1050	**0.1093**	4.10%	0.1894	**0.1895**	0.05%
ft7d	0.1455	0.1664	14.36%	0.0877	0.0949	8.21%	0.1751	0.1824	4.17%
ft8d	0.2056	0.2288	11.28%	0.0811	0.0832	2.59%	0.2009	0.2183	8.66%
la7d	0.1214	0.1411	16.23%	0.0759	0.0811	6.85%	0.1668	0.1724	3.36%
la8d	0.1152	0.1268	10.07%	0.0585	0.0630	7.69%	0.1422	0.1435	0.91%
avg	0.1469	**0.1658**	12.83%	0.0758	**0.0806**	6.27%	0.1713	**0.1792**	4.61%

Table 5: Performances measured by precisions at 10 documents

Exp	W1	W1+sense	ΔW1	W2	W2+sense	ΔW2	W3	W3+sense	ΔW3
ft7t	0.1500	0.1740	16.00%	0.1100	0.1120	1.82%	0.2340	0.2240	-4.27%
ft8t	0.1680	0.1840	9.52%	0.1760	0.1840	4.55%	0.3020	0.2920	-3.31%
la7t	0.1340	0.1540	14.93%	0.1440	0.1560	8.33%	0.2760	0.2820	2.17%
la8t	0.1340	0.1340	0.00%	0.1120	0.1060	-5.36%	0.2100	0.2020	-3.81%
avg	0.1465	**0.1615**	10.24%	0.1355	**0.1395**	2.95%	**0.2555**	0.2500	-2.15%
ft7d	0.1880	0.2200	17.02%	0.0940	0.1020	8.51%	0.2200	0.2140	-2.73%
ft8d	0.1740	0.1940	11.49%	0.1480	0.1500	1.35%	0.2800	0.2920	4.29%
la7d	0.1800	0.2120	17.78%	0.1060	0.1100	3.77%	0.2600	0.2800	7.69%
la8d	0.1420	0.1520	7.04%	0.1000	0.1060	6.00%	0.1880	0.2000	6.38%
avg	0.1710	**0.1945**	13.74%	0.1120	**0.1170**	4.46%	0.2370	**0.2465**	4.01%

Our approach is also successful in W2 and W3 experiments, but the degree of improvements were less compared to W1, especially in the W3 experiment. In the W3 experiment, the root sense tagging even deteriorates the retrieval performance on `la8t`. Since the result of W1 experiment on the `la8t` shows a good performance, it seems that the tf factor used in W2 and W3 caused some problems with our weight transforming method. We have found that W2+sense and W3+sense sometimes considerably raises or drops the term weights for the highly frequent terms in a document. When the sense fields for a highly frequent term in a query and in a document matches (i.e. sense-matching function q returns 1), sense weight 1.5 is multiplied to the original term weight, and there are more increases in the original term weights compared to the low frequency terms. Moreover the highly frequent terms have more possibilities to match with the senses of the query terms because of multiple-sense assignment within a document. This explains why the improvements in W2+sense and W3+sense are rather smaller than those for W1+sense. We think that the proper length normalization technique as well as term frequency modeling considering word senses should be developed for a more successful sense-based IR system.

Table 5 shows the precisions at 10 documents(p@10). In these results, we observe that there is a 10.24% improvement for short queries and 13.74% improvement for the long queries in W1 experiments, which are more improvements than those obtained by 11-point average precision measures. W1+sense certainly contributes to pulling up more relevant documents to the top ranks. In addition, W1+sense is more effective in long queries than in short queries, similar to the result in Table 4. It is easily imagined that more consistent sense tagging in a query is possible in long queries containing more context words for disambiguating each word. However, there is less improvement in W2 and W3 experiments, and W3+sense on title queries drops the baseline performances without the sense-based ranking method from 0.2555 to 0.2500. We think that this is due to the same overgrown weight problem.

Table 6: The number of improved and deteriorated queries by "W1+sense"

	impr.	detr.	same	sum
ft7t	28	9	13	50
ft8t	23	11	16	50
la7t	29	10	11	50
la8t	26	7	17	50
ft7d	28	13	9	50
ft8d	33	11	6	50
la7d	35	12	3	50
la8d	32	12	6	50

Table 6 shows the number of improved and deteriorated queries by W1+sense in average precisions. In all the experiments, more than half of the queries benefits from our root sense tagging method, although a number of queries become deteriorated. Since this is mainly due to inaccurate sense tagging results, more elaborate root sense tagging method must be developed for those queries.

4.3 Pseudo Relevance Feedback

A pseudo relevance feedback itself is a kind of method for semantic information retrieval since it aims at retrieving documents not directly containing query terms, but surely relevant to a given user query. We have conducted the pseudo relevance feedback experiments to investigate whether our root sense tagging approach can be used with relevance feedback. In this experiment, we have selected five terms from the top ten documents by the probabilistic term selection method suggested in [6], and added them to the original

Table 7: Performances with pseudo relevance feedback (adding 5 terms from the top 10 docs)

Exp	W1	W1+sense	ΔW1	W2	W2+sense	ΔW2	W3	W3+sense	ΔW3
ft7t	0.1521	0.1587	4.34%	0.0806	0.1032	28.04%	0.1830	0.2006	9.62%
ft8t	0.2115	0.2194	3.74%	0.1209	0.1259	4.14%	0.2303	0.2268	-1.52%
la7t	0.1458	0.1455	-0.21%	0.0942	0.1045	10.93%	0.1873	0.1930	3.04%
la8t	0.1322	0.1346	1.82%	0.0830	0.0827	-0.36%	0.1742	0.1831	5.11%
avg	0.1604	**0.1646**	2.59%	0.0947	**0.1041**	9.93%	0.1937	**0.2009**	3.70%
ft7d	0.1471	0.1813	23.25%	0.0733	0.0913	24.56%	0.1549	0.1830	18.14%
ft8d	0.2065	0.2342	13.41%	0.0906	0.0905	-0.11%	0.2097	0.2111	0.67%
la7d	0.1398	0.1633	16.81%	0.0787	0.0853	8.39%	0.1695	0.1856	9.50%
la8d	0.1245	0.1392	11.81%	0.0584	0.0594	1.71%	0.1440	0.1566	8.75%
avg	0.1545	**0.1795**	16.20%	0.0753	**0.0816**	8.47%	0.1695	**0.1841**	8.58%

query. For the sense fields of the new query terms in `+sense` experiments, we used a voting method that is the most frequent root sense in the top 10 documents is assigned to the terms to add.

Table 7 shows the 11-point average precision performances. As expected, further improvements are achieved by pseudo relevance feedback. Since W1+sense showed better precisions at top 10 documents in the previous experiments, it is obvious that the top 10 documents retrieved by a W1+sense initial search are more favorable sources to extract feedback queries than the documents retrieved by a W1 initial search. One interesting result is that W3+sense also achieves good performance improvements even though their initial search performances measured by p@10 are poor. We surmise that good feedback terms are selected due to the high initial search performance of W3+sense (0.2465), and the high quality of the set of added new query terms with their senses affects the performance.

From the performance improvements by relevance feedback with our root sense tagging approach, we can claim that even our crude sense assignment to the feedback query terms as well as sense-oriented term weighting are effective with pseudo relevance feedback since the initial search can locate more relevant documents near the top of the ranked list. However further experiments and analysis are needed.

4.4 BM25 using Root Senses

Figure 3 shows the experimental results with Okapi BM25 ranking function[3] considering word senses. We conducted this experiment to verify whether our proposed approach is appropriate for the existing state-of-the-art retrieval model. Our sense weight sw_{ij} is multiplied to the weight calculated by the BM25 formula. In this experiment, we evaluate the initial and feedback performances according to sense weight parameter α, unlike to the previous experiments where α is 0.5. The two collections are merged into one huge collection containing 337,899 documents, and we experimented on the collection. Needless to say, the baseline performance without sense information is the one where α is 0.

With this figure, we can observe that the initial search result is stable, while feedback performances with long query experiments fluctuate. We can achieve the best performance in all initial searches when α is approximately 0.5. There must be an appropriate α in the initial search, but not in the case of relevance feedback.

We think that the feedback performance is mainly affected by the quality of the set of new keywords used in feedback, not the quality of sense tagging. Among the selected feedback keywords, a few good keywords and their senses certainly contribute to retrieving more relevant documents. If there are, however, a number of bad keywords containing their senses, the performance significantly deteriorates by the feedback terms with the growth of the sense weight parameter. This is because the feedback performances fluctuate according to α.

Although some inconsistencies result from BM25 with the relevance feedback experiments, it is somewhat surprising that even our crude term weight transformation heuristics improve the BM25 performances in initial searches. We are confident that we can obtain better results by adopting our root sense tagging approach to any state-of-the-art retrieval model including BM25 if we develop more novel term weighting function considering word senses.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a coarse-grained, consistent, and flexible sense tagging method to improve large-scale text retrieval performance. For coarse-grained disambiguation, we have used only 25 unique beginner senses in WordNet instead of utilizing a large number of fine-grained senses. Thus, our approach can be applied to retrieval systems in other languages in cases where there are lexical resources much more roughly constructed than expensive resources like WordNet. Our sense tagger can be built without a sense-tagged corpus, and performs consistent disambiguation by considering only the single most informative neighboring word as evidence of determining the sense of target word. Multiple-sense assignment has been allowed so that the system can make the risks from disambiguation errors as small as possible. Although we added additional sense information to the retrieval system, the proposed sense-field based indexing and sense-weight oriented ranking do not seriously increase system overhead.

While many previous works on information retrieval using word senses often failed to improve retrieval effectiveness even in small text collections, our large-scale experiments on the TREC collection gave us promising results. More specifically, the *idf*-only term weighting experiment excluding the effect of term frequencies showed that our root sense tagger can find relevant documents more accurately. Other experiments with term frequencies also show good performance even with the relevance feedback method or state-of-the-art BM25 retrieval model, but we feel keenly the necessity for the elaborate term weighting method considering word senses.

For future work, we will focus on the following two prob-

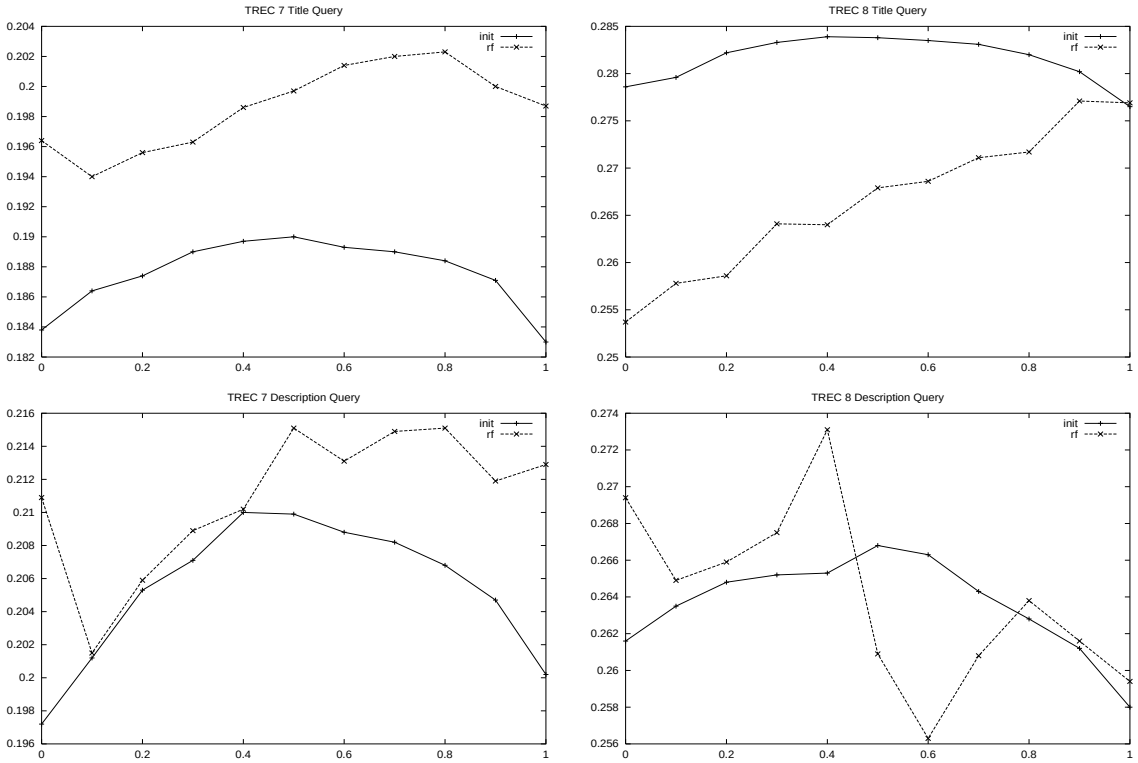

Figure 3: Initial and Feedback Performances of BM25+sense according to sense weight parameter α

lems that we have encountered through the experiments. First, we realize that verbs also should be assigned with senses for further improvement because the words used in noun form within a query are often used in verb form in the relevant documents. Since the verbs are not sense-tagged in our work, our system often fails to match their senses. Second, it is essential to develop an elaborate retrieval model, i.e., a term weighting model considering word senses. Such a model may be an entirely new one or developed based on existing term-based retrieval models.

6. REFERENCES

[1] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[2] T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999.

[3] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1. *Information Processing and Management*, 36(6):779–808, 2000.

[4] R. Krovetz and W. B. Croft. Lexical ambiguity and information retrieval. *Information Systems*, 10(2):115–141, 1992.

[5] C. D. Manning and H. Schutze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[6] S. E. Robertson and S. Walker. Okapi/keenbow at trec-8. In *Proceedings of TREC-8, 8th Text Retrieval Conference*, pages 151–161, Gaithersburg, US, 2000.

[7] M. Sanderson. Retrieving with good sense. *Inf. Retr.*, 2(1):49–69, 2000.

[8] M. Sanderson and C. J. V. Rijsbergen. The impact on retrieval effectiveness of skewed frequency distributions. *ACM Transactions on Information Systems*, 17(4):440–465, 1999.

[9] H. Schutze and J. Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1995.

[10] C. Stokoe, M. P. Oakes, and J. Tait. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 159–166. ACM Press, 2003.

[11] E. M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180. ACM Press, 1993.

[12] P. Wallis. Information retrieval based on paraphrase. In *Proceedings of the 1st Pacific Association for Computational Linguistics Conference*, 1993.

[13] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.